A science-gateway workload archive application to the self-healing of workflow incidents

Rafael Ferreira da Silva, Tristan Glatard, Frédéric Desprez

A science-gateway workload archive application to the self-healing of workflow incidents

Rafael Ferreira da Silva (1), Tristan Glatard (2), Frédéric Desprez (3)
(1) rafael.silva@creatis.insa-lyon.fr, (2) glatard@creatis.insa-lyon.fr
Université de Lyon, CNRS, INSERM, CREATIS, Villeurbanne, France
(3) Frederic.Desprez@inria.fr
INRIA, Université de Lyon, LIP, ENS Lyon, Lyon, France

Overview

Information about the execution of distributed workload is important for studies in computer science and engineering, but workloads acquired at the infrastructure-level reputably lack information about users and application-level middleware. Meanwhile, workloads acquired at science-gateway level contain detailed information about users, pilot jobs, task sub-steps, bag of tasks and workflow executions. In this work, we present a science-gateway archive, we illustrate its possibilities on a few case studies, and we use it for the autonomic handling of workflow incidents.

Introduction, challenges

Execution logs are widely used for research on distributed systems, to validate assumptions, model computational activity, and evaluate methods in simulation or in experimental conditions. Some of these logs are collected by workload archives, such as the Grid Workloads Archive [1], and made available online to the community. However, as they are gathered at infrastructure level, they lack critical information about dependencies among tasks, about task sub-steps, about artifacts introduced by application-level scheduling, and about users. On the other hand, information acquired at science-gateway level adds value to several case studies related to user accounting, pilot jobs, fine-grained task analysis, bag of tasks, and workflows.

In this work, we present our science-gateway archive extracted from the Virtual Imaging Platform (VIP) [2] and available in the Grid Observatory [3], we illustrate its possibilities on a few case studies, and we show how it can be used for the autonomic handling of workflow incidents.

A science-gateway archive

Our science-gateway archive is based on the workload of the Virtual Imaging Platform. VIP is an open web platform for medical simulation. Users authenticate to a web portal with login and password, and are then mapped to X.509 robot credentials. Applications deployed in VIP are described as workflows executed using MOTEUR workflow engine [4]. Resource provisioning and task scheduling is provided by DIRAC [5] using pilot-jobs. Tasks are executed on the biomed virtual organization of the European Grid Infrastructure (EGI)[1].

Our science-gateway archive model adopts the schema on Figure 1. *Task* contains information such as final status, exit code, timestamps of internal steps, application and workflow activity name. Each task is associated to a *Pilot Job*. *Workflow Execution* gathers all the activities and tasks of a workflow execution, *Site* connects pilots and tasks to a grid site, and *File* provides the list of files associated to a task and workflow execution. In this work we focus on *Task*, *Workflow Execution* and *Pilot Job*.

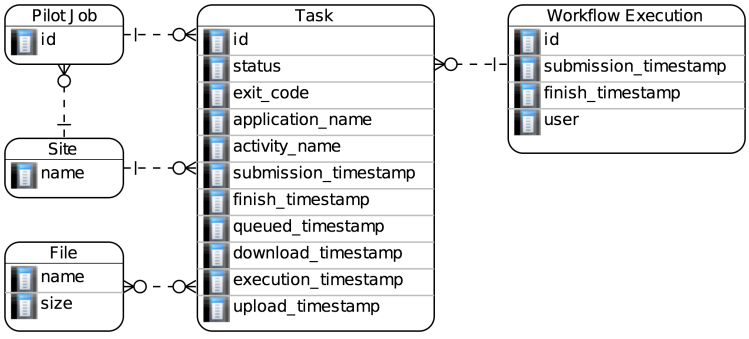

Figure 1. Science-gateway archive model.

The science-gateway archive is extracted from VIP. *Task, Site* and *Workflow Execution* information are acquired from databases populated by the workflow engine at runtime. *File* and *Pilot Job* information are extracted from the parsing of task standard output and error files.

Studies presented in this work are based on the workload of VIP from January 2011 to April 2012. It consists of 2,941

[1] http://www.egi.eu

workflows executions, 112 users, 339,545 pilot jobs, 680,988 tasks where 338,989 are completed tasks, 138,480 error tasks, 105,488 aborted tasks, 15,576 aborted task replicas, 48,293 stalled tasks and 34,162 submitted or queued tasks. Stalled tasks are tasks which lost communication with the pilot manager, e.g. because they were killed by computing sites due to quota violation. Traces used in this work are available to the community in the Grid Observatory[2].

Pilot jobs are increasingly used to improve scheduling and reliability on production grids. This type of workload, however, is difficult to analyze from infrastructure traces as a single pilot can wrap several tasks and users without informing the infrastructure. Figure 2 shows the number of tasks and users per pilot presented in the archive. Most pilots (83%) execute only 1 task due to walltime limits or other discards, which represents 62% of the task set. On workloads acquired at infrastructure level, about 38% of the task set would be misunderstood. The distribution of users per pilots has a similar decrease: 95% of the pilots execute tasks of a single user.

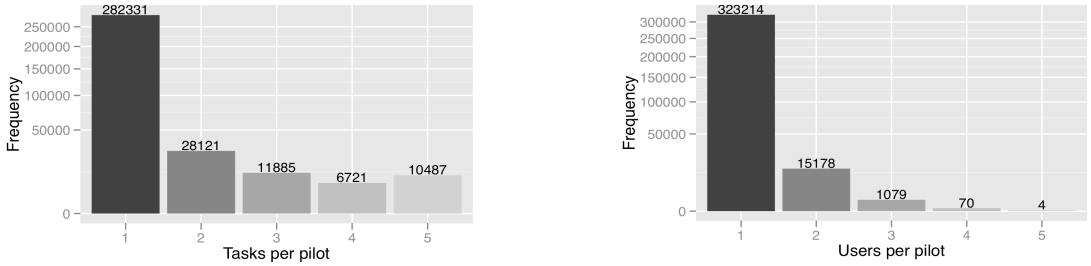

Figure 2. Histogram of tasks per pilot (left) and users per pilot (right).

Figure 3 compares the number of submitted jobs, consumed CPU time, and consumed wall-clock time obtained by the EGI infrastructure and by VIP. The number of jobs reported by EGI is almost twice as important as in VIP. This huge discrepancy is explained by the fact that many pilot jobs do not register to the pilot system due to some technical issues, or do not execute any task due to the absence of workload, or execute tasks for which no standard output containing the pilot id could be retrieved. These pilots cannot be identified from infrastructure-level logs. It also reveals that a significant fraction of the workload measured by EGI does not come from applications but from artifacts introduced by pilot managers.

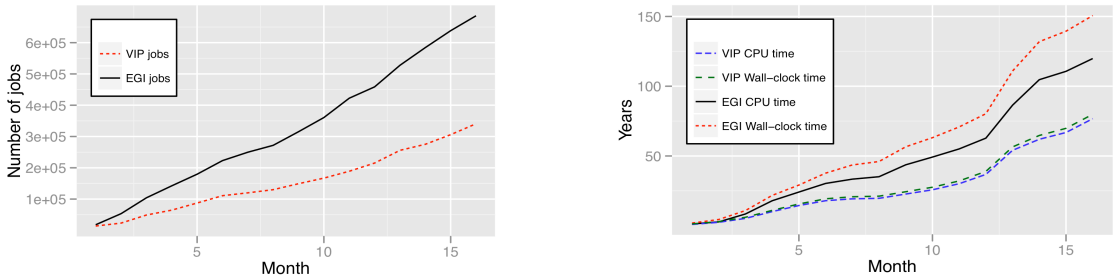

Figure 3. Number of submitted pilot jobs (left), and consumed CPU and wall-clock time (right) by the infrastructure (EGI) and the science gateway (VIP).

Traces acquired at the science-gateway level provide fine-grained information about tasks, which is usually not possible at the infrastructure level. Figure 4 (left) shows the distributions of download, upload and execution times for successfully completed tasks. Distributions show a substantial amount of very long steps. Figure 3 (right) shows the occurrence of 6 task-level errors. These error codes are application-specific and not accessible to infrastructure level archives.

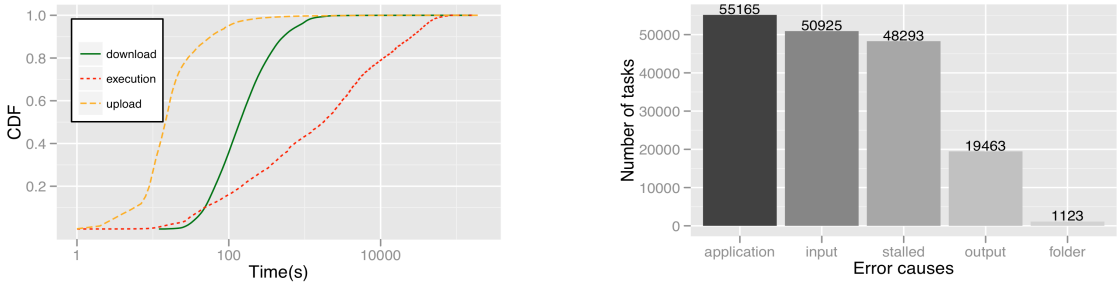

Figure 4. Different steps in task life (left) and task error causes (right).

Last, bag-of-tasks (BoT) can be properly identified from the science-gateway level, which is not the case at the infrastructure level. For instance, [6] detects bags of tasks as tasks submitted by a single user in a given time interval. However, this

[2] http://www.grid-observatory.org

method is very inaccurate and its use may have important consequences on works based on such detection. Figure 5 presents the comparison of BoT characteristics obtained from the described method and VIP. BoTs in VIP were extracted as the tasks generated by the same activity in a workflow execution and are considered as ground truth (named *Real Non-Batch* for single BoTs and *Real Batch* for others). Analogously, we name *Non-Batch* and *Batch* BoTs determined by the method. The method considers that 90% of BoT sizes ranges from 2 to 10 while these batches represents about 50% of *Real Batch*. Moreover, single BoTs are overestimated up to 400% on their durations. Both, single and batches, are underestimated by about 30% on inter-arrival times and up to 25% on consumed CPU times.

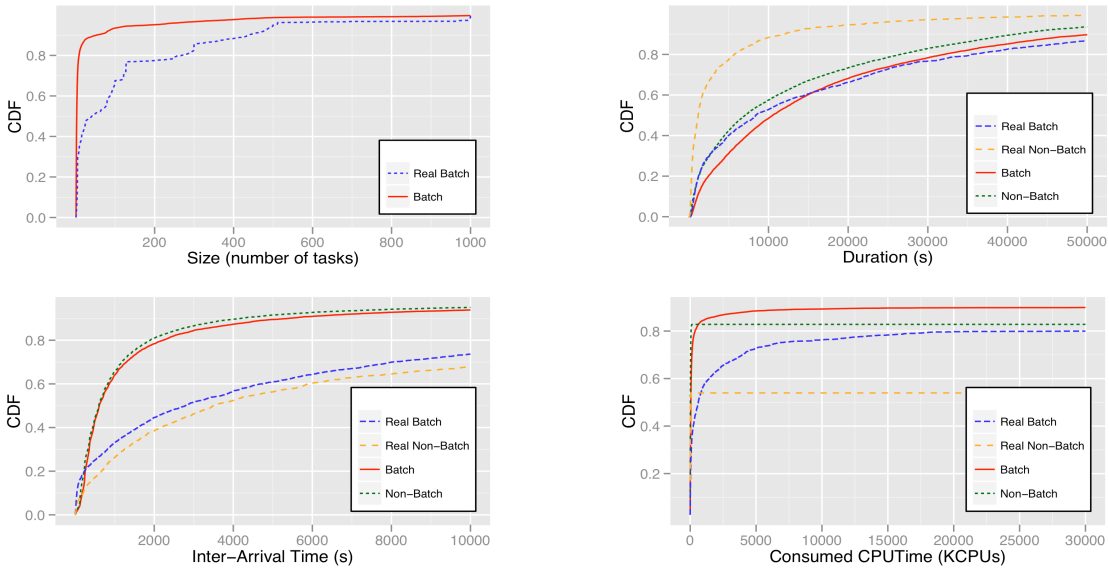

Figure 5. Cumulative Distributed Function (CDF) of characteristics of batched and non-batched submissions: BoT sizes, duration per BoT, inter-arrival time and consumed CPU time.

Application to workflow self-healing

Distributed computing infrastructures (DCI) are becoming daily instruments of scientific research, in particular scientific gateways [7] developed to allow scientists to transparently run their analyses on large sets of computing resources. Operating these gateways require an important human intervention to handle operational incidents. One strategy to cope with these incidents is to automate such operations by using a healing process that could autonomously detect and handle them. The self-healing process presented in [8] handles incidents by automating such operations through a MAPE-K loop: monitoring, analysis, planning, execution and knowledge.

Figure 6 presents the MAPE-K loop of the healing process. The monitoring phase consists in collecting events (job completion and failures) or timeouts. In the analysis phase, incident degrees and levels are determined from events and execution logs (historical information) extracted from the science-gateway archive. Then, an incident is selected by using a combination of roulette wheel selection and association rules (planning phase). Roulette wheel selection assigns a proportion of the wheel to each incident level according to their probability and a random selection is performed based on a spin of the roulette wheel. Association rules are used to identify relations between levels of different incidents based on historical information. Finally, a set of actions is performed to handle the selected incident.

Incident level numbers and thresholds values are set from visual mode detection from histograms based on error causes and task sub-steps durations. The healing process was implemented in the Virtual Imaging Platform and deployed in production. It uses a subset from the science-gateway archive as historical information knowledge. Results presented hereafter show the ability of the healing process to improve workflow makespan in case of recoverable incidents and quickly identify and report critical issues.

Experiments were conducted on two workflow activities: *FIELD-II/pasa* and *Mean-Shift/hs3*. The former consists of 122 invocations of an ultrasonic simulator on an echocardiography 2D data set. It is a data-intensive activity that transfers about 210 MB of data and consumes up to 15 minutes of CPU time. The last has 250 CPU-intensive application invocations of an image filtering application that consumes up to 1 hour and transfers about 182 MB of data.

Figure 7 shows the makespan of both applications for a correct execution where all the input files exist and the application is supposed to run properly and produce the expected results. The makespan was considerably reduced in all repetitions with speed-up values ranged from 1.3 to 4.

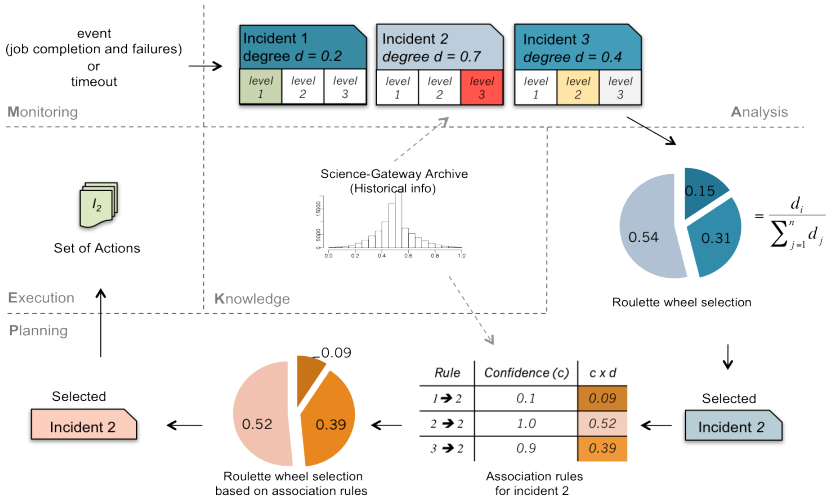

Figure 6. MAPE-K loop of the healing process.

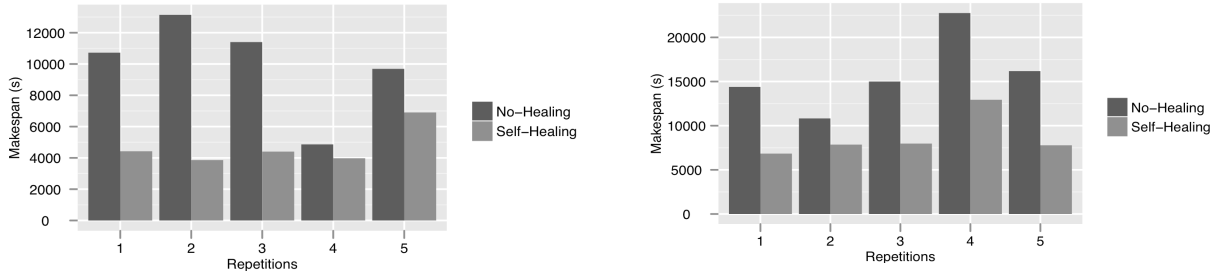

Figure 7. Execution makespan for *FIELD-II/pasa* (left) and *Mean-Shift/hs3* (right) for a correct execution.

Conclusion

We presented a science-gateway workload archive containing detailed information about users, pilot jobs, tasks and bag of tasks. First, we illustrated the added value of science-gateway workloads compared to infrastructure-level traces using information collected by the Virtual Imaging Platform in 2011/2012. Second, we showed an applicability of the workload archive on a healing process to cope with workflow execution incidents by quantifying them through historical execution information. Results show that the self-healing method proposed in [8] speeds up execution up to a factor of 4.

Traces acquired by the Virtual Imaging Platform will be regularly made available to the community in the Grid Observatory. We hope that other science-gateway providers could also start publishing their traces so that computer-science studies can better investigate production conditions.

References

[1] A. Iosup, H. Li, M. Jan, S. Anoep, C. Dumitrescu, L. Wolters, D.H.J. Epema, "The grid workloads archive", in *Future Generation Computer Systems*, 24(7) (2008) 672-686

[2] R. Ferreira da Silva, S. Camarasu-Pop, B. Grenier, V. Hamar, D. Manset, J. Montagnat, J. Revillard, J. R. Balderrama, A. Tsaregorodtsev, T. Glatard, "Multi-Infrastructure Workflow Execution for Medical Imaging Simulation in the Virtual Imaging Platform", in *HealthGrid Conference 2011*, Bristol, UK (2011)

[3] C. Germain-Renaud, A. Cady, P. Gauron, M. Jouvin, C. Loomis, J. Martyniak, J. Nauroy, G. Philippon, M. Sebag, "The grid observatory", in *IEEE International Symposium on Cluster Computing and the Grid*, (2011) 114-123

[4] T. Glatard, J. Montagnat, D. Lingrand, X. Pennec, "Flexible and Efficient Workflow Deployment of Data-Intensive Applications on Grids with MOTEUR", in *International Journal of High Performance Computing Applications (IJHPCA)*, 22(3) (2008) 347-360

[5] A. Tsaregorodtsev, N. Brook, A. C. Ramo, P. Charpentier, J. Closier, G. Cowan, R. G. Diaz, E. Lanciotti, Z. Mathe, R. Nandakumar, S. Paterson, V. Romanovsky, R. Santinelli, M. Sapunov, A. C. Smith, M. S. Miguelez, A. Zhelezov, "DIRAC 3. The New Generation of the LHCb Grid Software", in *Journal of Physics: Conference Series*, 219(6) (2009)

[6] A. Iosup, M. Jan, O. Sonmez, D. Epema, "The characteristics and performance of groups of jobs in grid", in *Euro-Par* (2007) 382-393

[7] S. Gesing, J. van Hemert, "Concurrency and Computation: Practice and Experience", in *Special Issue on International Workshop on Portal for Life-Sciences 2009*, 23(3) (2011)

[8] R. Ferreira da Silva, T. Glatard, F. Desprez, "Self-healing of operational workflow incidents on distributed computing infrastructures", in *12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid),* Ottawa, Canada (2012) 318-325